REMARKS

Claims 1-46 are pending. Claims 1-46 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Windows 2000 Authentication (http://www.comptechdoc.org/os/windows/win2k/win2kauthentication.html) ("Windows").

Reconsideration is requested. The rejections are traversed. No new matter is added. Claim 47 is added. Claims 1-47 remain in the case for consideration.

REJECTIONS UNDER 35 U.S.C. § 103(a)

In rejecting the claims, the Examiner has made several arguments. These arguments fall into two categories:

- "Windows" makes the claimed invention obvious.
- "Windows" is proper prior art.

The Applicant also notes that, in arguing that "Windows" is proper prior art unless the Applicant can prove otherwise, the Examiner has essentially asked the Applicant to perform a search. This is an attempt by the Examiner to shift the burden of examination onto the Applicant. The Applicant suggests that this burden shifting is inappropriate, as argued below. Nevertheless, the Applicant has conducted a search in an attempt to determine exactly what Microsoft Windows 2000 offered in terms of authentication as of its initial release. On the premise that if anyone had published anything about authentication in Microsoft Windows 2000 before the filing date of this patent application, the Applicant searched the knowledge base of the website of Microsoft Corporation for documents relating to authentication in Windows 2000. The Applicant's search turned up one publication by Microsoft: "Basic Overview of Kerberos User Authentication Protocol in Windows 2000" (http://support.microsoft.com/?kbid=217098) ("Kerberos"), a copy of which is attached. This reference is not proper prior art any more than "Windows" is proper prior art: no date of publication for "Kerberos" is provided, but "Kerberos" was last "reviewed" on November 21, 2003, and has a copyright date of 2006. "Kerberos" provides more detail about how Microsoft Windows 2000 performs authentication, and shows that some of the Examiner's interpretations of "Windows" are erroneous, as discussed below.

The Applicant addresses the Examiner's arguments in both of these categories in turn.

- "Windows" does not make the claimed invention obvious

In this section, the question as to whether "Windows" is proper prior art is ignored. In other words, even though the Applicant argues that "Windows" is not proper prior art (see below), the Applicant argues herein why the claimed invention is not obvious from "Windows".

The Examiner's omnibus rejection of claims 4-18 and 23-46 is inappropriate

First, the Applicant notes that the Examiner has only provided a specific rejection of claims 1-3 and 19-22. The Examiner has made a summary rejection of claims 4-18 and 23-46, stating that "such particular features are well known in the art for the purposes of handling information across computers" (*see* Office Action dated April 21, 2005, page 5). The Applicant respectfully disagrees. The Applicant asserts that these claims individually include features that are novel and non-obvious.

For example, claims 38-42 all recite a federation access policy. Notwithstanding the Examiner's statement "it was well known in the art to have a 'federated' situation among multiple computers that are networked and controlled with domain controllers, especially in domain controller that have group policy information which is replicated to all domain controllers" (*see* Office Action dated April 21, 2005, page 3), the Applicant respectfully asserts that a federation access policy was not known in the art at the time of invention. In particular, the Applicant asserts that a federation access policy used when two computers were in different domains was not known in the art. The Applicant points out that, for example, claim 1 recites a "cross-domain authentication apparatus", and "a first computer on a first domain and a second computer on a second domain". In other words, the domains in the claimed invention are different. This point is emphasized in new claim 47. The Examiner insists that it was known to have replicated domain information; but if the domains in the claimed invention are different, then the Examiner's assertion that a "'federated' situation" was well known is off point: the Examiner needs to show that it was known in the art to have a "'federated' situation" with different, non-replicated domains.

Further, M.P.E.P. § 707.07(d) instructs the Examiner to provide specific rejections of the individual claims. M.P.E.P. § 707.07(d) states that "omnibus rejection of the claim 'on the references and for the reasons of record' is stereotyped and usually not informative and should therefore be avoided. This is especially true where certain claims have been rejected on one

ground and other claims on another ground. A plurality of claims should never be grouped together in a common rejection, unless that rejection is equally applicable to all claims in the group." Thus, the M.P.E.P. instructs the Examiner to provide a specific rejection of the claims over the art. If the Examiner's assertion that "such particular features are well known in the art", then the Examiner should be able to provide one or more references that show that the features were "well known in the art"; the Examiner's cursory rejection of the claims is inappropriate.

"Windows" does not teach a shared secret as claimed

The following is taken from the Response to the Office Action dated September 8, 2004:

> Further problems exist within the Examiner's reasoning, even ignoring all of the other problems with Windows as a reference. The Examiner asserts that the password used in a domain logon is a "shared secret". The Applicant disagrees. The shared secret of the invention is a secret known in advance to each of the computers (*see* Specification, page 3, lines 3-4). But in a domain logon, the password is not known to the local computer until provided by the user. (If the password were known to the local computer, then the logon would be a local logon, and not a domain logon.) Therefore, the password cannot be a secret that is known in advance by each computer.
>
> Another reason the password in Windows cannot be the shared secret is that the password "is sent to the Windows 2000 domain controller with an authentication request" (*see* "Windows", "Process of Logging On", ¶ 2). The mere fact that the password is being sent to the domain controller establishes that the password is not a shared secret. Consider the situation where an unauthorized third party is attempting to access resources. Assuming the third party does not know the correct password, he would have to provide an incorrect password. But if the domain controller and the local computer "know" different passwords, the secret is clearly not shared.
>
> It is also worth noting that step 325, wherein the mediator performs the authentication, does not mention the shared secret. Instead, the shared secret is mentioned in step 340, wherein the random challenge nonce is encrypted using the shared secret. This shows that the user's password, which is used for user

authentication, is different from the shared secret. If the password and the shared secret were meant to be the same element, the specification would have said so.

It could happen that the shared secret happens to be identical to the user's password. But if this situation arises, it is only by remote coincidence. As described in the specification, the user also has to provide his password to authenticate himself (if the local machine has not stored the authentication in advance).

In response, the Examiner states that "[a] shared secret is a secret that is shared. A password can be a secret. How can a password not be a secret?" (*see* Office Action dated April 21, 2005, page 2). The Applicant readily agrees that a password can be a secret. But the Examiner has overlooked, among other things, the significance of the word "shared": the fact that something is "secret" does not automatically mean that the secret is "shared".

It would appear that the Examiner is relying on the language in "Windows" that "the name and password are checked against the local database" (*see* "Windows", page 1). The Examiner's reasoning would appear to be as follows: "A password is a secret. The database knows the password, as (intuitively) does the user, so the password is shared. Therefore, the password is a secret shared, which meets the language of the claim."

In following this logic, the Examiner has overlooked the fact that the name and password are checked against the local database "[i]f the logon is local" (*see Id.*). A "local login" is where the user is logging into the local machine only: in this situation, there is no logon to the domain. That this situation excludes the possibility of a logon to a domain can be seen not only from the use of the term "local", but also from the next sentence in "Windows", which begins "[i]f the logon is a domain logon . . ." (*see Id.*). The author of "Windows" is contrasting the situation where the logon is to the local machine only against the situation where the logon is to a domain. If the logon is local, it is not a domain logon.

Having established that the logon in question in the first sentence of ¶ 2 of "Process of Logging On" in "Windows" is a local logon, it should now be clear that the password cannot be a "secret shared" as claimed. Claim 1 recites "a secret shared between the first and second computers". In the local logon situation, the password is known only by the user and the local computer. Since the user is neither the "first computer" nor the "second computer", the password is not "a secret shared between the first and second computers". Thus, in the situation

where the logon is local, a password is not a "secret shared" as claimed, and claims 1 and 27, both of which recite a "secret shared", are neither anticipated nor made obvious by "Windows".

There is a possibility that the Examiner was arguing that the password was a shared secret in the situation where the logon is a domain logon. In the domain logon situation, there are two computers. But in this situation, the two computers do not share the password. This can be seen from the fact that ¶¶ 3-4 of "Process of Logging On" in "Windows" does not mention comparing the password received from the user with a database. Instead, the password and user name are "encrypted into a key" (*see Id.*).

The Examiner might argue that "encryption" suggests that the user name and password can be decrypted. The Applicant respectfully points out that this is not specifically stated anywhere in "Windows": the Examiner would therefore be drawing an unsupported inference. Further, the Applicant asserts that this decryption not only does not happen, but in fact cannot occur. The explanation why is below.

First, the Applicant points the Examiner to the first sentence of "The CTDP Windows 2000 Tutorial Version 0.6.1 Oct. 28, 2001" (http://www.comptechdoc.org/os/windows/win2k/index.html) ("Introduction"), which is the introduction to the electronic document that includes "Windows". A copy of "Introduction" is attached to this response. According to the first sentence of "Introduction", "[t]his guide may have inaccuracies, use at your own risk". The author of "Windows" has explicitly acknowledged that nothing he or she says is guaranteed to be accurate. Thus, where there is reason to believe the author is incorrect, the Applicant believes the author's statements should not be trusted. Given how cursory an explanation of domain logon the author of "Windows" provides, the Applicant suggests the author of "Windows" is glossing over many important details.

The Applicant points the Examiner to "Kerberos", mentioned above. The Applicant again points out that "Kerberos" is not proper prior art any more than "Windows" is proper prior art. But as "Kerberos" is authored by Microsoft Corporation, the company that developed, manufactured, and sold Microsoft Windows 2000, the Applicant respectfully submits that there is no better person or corporate entity capable of describing how Microsoft Windows 2000 performs authentication than Microsoft Corporation, its developer. Therefore, Microsoft Corporation is a more reliable source as to how Microsoft Windows 2000 operates than the author of "Windows". According to "Kerberos":

b. The Kerberos client sends a message to the Key Distribution Server (KDC), of type KRB_AS_REQ (Kerberos Authentication Server Request). This message has two parts:

- An identification of the user, A, and the service for which she is requesting credentials, the TGS (Ticket-Granting Service).
- Pre-authentication data, intended to prove that A knows her password. This is simply an authenticator encrypted with A's master key. The master key is generated by running A's password through a OWF.

c. The KDC, upon receipt of KRB_AS_REQ from A, looks up the user A in its database (the Active Directory), gets her master key, decrypts the pre-authentication data, and evaluates the time stamp inside. If the time stamp passes the test, the KDC can be assured that the pre-authentication data was encrypted with A's master key, and is not merely a captured replay.

(*see* "Kerberos", page 1)

Given what is described in step c. about looking up user A in the database, this lookup would be performed based on the identification of the user, in the first part of the message sent to the KDC: after all, a password does not identify a user (using a password to identify a user would require that no two users share the same password, and rejecting a password as a duplicate would reveal that some user is using that password, information that is better kept secret). So the identification data does not include the password. This is reinforced by the second part of the message indicating that the pre-authentication data is used to prove that A knows her password (that is, to verify A's identity).

But neither does the pre-authentication data include the password. Note that the pre-authentication data is "simply an authenticator encrypted with A's master key" (*see Id.*). Further, according to step c., the "authenticator" includes a time stamp (presumably, of when the user attempted to log on, to protect against an intruder providing a copy of an earlier logon attempt). Given that there is separate mention of the "authenticator", the "master key", and the "password", it seems reasonable to conclude that these three items are distinct.

So if the authenticator is not the password, is the master key the password? No, it is not, as the master key is described as "generated by running A's password through a OWF" (*see Id.*).

This, however, begs the question: what is a "OWF"? After all, if a "OWF" is an encryption algorithm, then the password might be recoverable by "decrypting" the master key.

The Applicant suggests that the acronym "OWF" stands for "one-way function". This is consistent with other uses of the acronym OWF by Microsoft Corporation. For example, the Applicant has attached a copy of "List of Security Fixes in Windows 2000 Service Pack 3" (http://support.microsoft.com/kb/324953/en-us) ("Security Fixes"). No date of publication for "Security Fixes" is provided; "Security Fixes" was last "reviewed" on March 9, 2006 and with a copyright date of 2006, and is therefore not proper prior art any more than either "Windows" or "Kerberos" is proper prior art. But "Security Fixes" is not being presented for its content: "Security Fixes" is being presented to show a use of the acronym "OWF" in the context of Microsoft Windows 2000, and the definition of the acronym "OWF".

Bullet point number 2 of "Security Fixes" reads "Set LAN Manager (LM) One-Way Function (OWF) Password Results in Access Denied Error" (*see* "Security Fixes", page 1). While neither "Kerberos" nor "Security Fixes" defines what is meant by a "one-way function", it is commonly understood that a one-way function is a function that is difficult to invert: that is, given a result of the one-way function, it is difficult to determine the input to the one-way function that produced that result. For example, "one-way function from FOLDOC" (http://foldoc.org/foldoc.cgi?query=one-way+function&action=Search) defines "one-way function" as a "function which is easy to compute but whose inverse is very difficult to compute. Such functions have important applications in cryptography, specifically in public-key cryptography" (*see* "one-way function from FOLDOC", page 1). Because one-way functions are difficult to invert, given a known result it is difficult to find any input data that would produce the result, even if more than one input might produce the same result (which is not guaranteed: in general, there is now way to know how many inputs might produce a particular output). Hash functions are excellent examples of one-way functions, in that the original data cannot be recovered from the result of the hash.

Returning to "Kerberos", the master key is thus the result of sending the user's password through the one-way function. Because a one-way function is "one-way", the original password cannot be recovered from the master key: that is, the master key cannot be "decrypted". (It is worth noting that, as one-way functions are hard to invert, and it is essentially impossible to find an input that returns a given result, an interloper would not be able to provide a password that

would hash to the master key.) Thus, even if the master key is stored in the clear (where anyone could read it: this is unlikely in Microsoft Windows 2000 because the master key is used for encryption and decryption), the master key does not provide any information about the user's password.

So the user's password is not part of the identification data or the pre-authentication data, and is not derivable from any of that data or the master key. What does that mean? First, it is worth noting that the only data stored by the KDC in its database is the master key (presumably indexed by the user's identification). But if the password cannot be derived from the master key, then the password is not known by the KDC. This means that at least one of the two computers involved in domain logon in Microsoft Windows 2000 does not know the password, which means the password is not a "shared" secret.

Thus, given that the KDC does not store the user's password (instead storing the master key, derived from the password using a one-way function), and that the "Kerberos" client does not store the password either, the password is not a "secret shared". The password is a secret, yes: but it is a secret even from the computers, and so cannot be "a secret shared between the first and second computers", as claimed.

As "Windows" does not teach or suggest a shared secret, whether using a local logon or a domain logon, claims 1 and 19 are patentable under 35 U.S.C. § 103(a) over "Windows". Claims 1 and 19 are therefore allowable, as are dependent claims 2-18 and 20-47.

"Windows" does not teach cross-domain authentication as claimed

In response to the Office Action dated September 8, 2004, the Applicant argued that "Windows" does not teach cross-domain authentication. In fact, "Windows" only describes local logon and domain logon. The Examiner did not address this argument at all in the Response to Arguments included in the Office Action dated April 21, 2005. M.P.E.P. § 707.07(f) states that "[w]here the applicant traverses any rejection, the examiner should, if he or she repeats the rejection, take note of the applicant's argument and answer the substance of it". As the Examiner made no response to the argument that "Windows" does not teach or suggest cross-domain authentication, the Examiner's Office Action is incomplete. For the Examiner's convenience, this argument is restated here:

Third, Windows does not disclose cross-domain identification. As stated clearly in the preamble of claims 1 and 19, the invention is directed toward an apparatus and method for cross-domain authentication. As described in the specification, cross-domain authentication is used where "an identity in one domain can be authenticated to another domain in the federation without actually creating an identity in the latter" (*see* Specification, page 2, lines 28-30). Windows on its face is limited to two situations: local logons and domain logons (*see* Windows, Process of Logging On, ¶ 2). Neither of these situations can handle a cross-domain authentication, where the computers involved are in different domains.

Although the Applicant believes that the preamble sufficiently describes claims 1 and 19 to distinguish the claims over Windows, claims 1 and 19 have been amended to explicitly identify the computers as being on different domains. The Applicant also asserts that, because Windows is not proper prior art and the amendments merely clarify features already present in the claims, the amendments to claim 1 and 19 are not narrowing amendments.

As "Windows" does not teach or suggest cross-domain authentication, claims 1 and 19 are patentable under 35 U.S.C. § 103(a) over "Windows". Claims 1 and 19 are therefore allowable, as are dependent claims 2-18 and 20-47.

"Windows" does not teach a "federated access policy" as claimed

In the Office Action dated September 8, 2004, the Examiner stated that "it was well known in the art to have a 'federated' situation among multiple computers that are networked and controlled with domain controllers – especially in domain controllers that have group policy information which is replicated to all domain controllers, such as in Windows 2000 SYSVOL that is noted at the last sentence of the "Windows" reference – for the motivation of having easier control of a group of domain controllers" (*see* Office Action dated September 8, 2004, page 3).

The Applicant asserts that whether or not a "federated situation" was known in the art, the Examiner has misread the claims. The claims describe a "federation access policy", which is distinguishable from a network of domain controllers. According to the specification, a

"federation access policy is used to specify rights authorization of local resources to any identity in the federated identity space" (*see* specification, page 13, lines 20-21; *see also, e.g.*, specification, page 2, lines 21-34 and page 30, line 18 through page 15, line 15). As a "federation access policy" specifies the rights authorization of local resources to an identity in the federated identity space" and a "federated situation" is a "federated network of computers", a "federated situation" has nothing to do with a "federation access policy", which the Applicant asserts is distinguishable. Thus, "Windows" does not teach or suggest "a federation access policy identifying access permission on the first computer on the first domain for a user local to the second computer on the second domain over the network", as claimed.

As "Windows" does not teach or suggest a federation access policy, claims 1 and 19 are patentable under 35 U.S.C. § 103(a) over "Windows". Claims 1 and 19 are therefore allowable, as are dependent claims 2-18 and 20-47.

In the above arguments, the Applicant has focused on claims 1 and 19, as these are the only claims the Examiner has rejected in any detail. The Examiner has rejected claims 2-18 and 20-46 as including features supposedly "known in the art" without providing any support for this assertion. The Applicant respectfully requests that the Examiner support his assertion that the features of claims 2-18 and 20-46 are "known in the art". The Applicant believes that claims 2-18 and 20-47 include features that are patentable independently of claims 1 and 19, and are therefore also allowable. The Applicant reserves the right to argue the independent patentability of claims 2-18 and 20-47, once the Examiner has supported his assertion that the features of these claims were "known in the art".

- "Windows" is not prior art under the M.P.E.P.

In this section, the Applicant argues again why "Windows" is not a proper reference to use in rejecting the claims, and that the Examiner's rejection is therefore improper.

"Windows" is not proper prior art under 35 U.S.C. § 103(a)

According to the Office Action dated June 9, 2006, the Applicant's argument that the reference cannot be considered valid prior art "is not persuasive because the Office cited the prior art as a teaching regarding the Microsoft Windows 2000 rather than the prior art being

published in the year 2000" (*see* Office Action dated June 9, 2006, page 2). This rationale expressly ignores both the M.P.E.P. and the statute, and should be reconsidered.

As the Examiner is rejecting the claimed invention over "Windows" under 35 U.S.C. § 103(a), the text of the statute is worth reviewing. According to 35 U.S.C. § 103(a), "[a] patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains". Thus, for a reference to be available under 35 U.S.C. § 103(a), the reference must first qualify as prior art under 35 U.S.C. § 102.

35 U.S.C. § 102 describes seven different reasons why a patent should not be granted. Of these seven, subsections (c), (d), and (f) relate to actions of the Applicant. As the Applicant has not done anything to justify denying the Applicant this patent, these subsections do not apply. Subsection (g) deals with interferences and conflicts between multiple patent applications to the same subject matter. As "Windows" is not another patent application, subsection (g) does not apply. Subsection (e) pertains to issued patents and published patent applications: clearly, "Windows" is neither of these items, so subsection (e) is not applicable. This leaves subsections (a) and (b). Both subsections (a) and (b) use dates to determine whether a reference can be used to reject a claim. Subsection (a) requires that the invention be "described in a printed publication in this or a foreign country, before the invention thereof by the applicant for patent": in other words, the date the reference was published must be at least before the filing date of the patent application. Subsection (b) requires that the invention be "described in a printed publication in this or a foreign country . . . more than one year prior to the date of the application for patent in the United States".

As previously argued, the earliest publication date for "Windows" that can be determined is October 28, 2001. This is several months after March 22, 2001, the filing date of the patent application. Accordingly, "Windows" is not available as a reference under either 35 U.S.C. § 102(a) or (b). But if "Windows" is not available as a reference under 35 U.S.C. § 102(a) and (b), and is not available under any of subsections (c)-(g), then "Windows" is not available at all under 35 U.S.C. § 102, and so is not available under 35 U.S.C. § 103(a) to argue that the claimed invention is obvious.

It is worth noting that 35 U.S.C. § 102(a) and (b) do not depend on the content of a reference; for purposes of deciding whether the reference is prior art. 35 U.S.C. § 102(a) and (b) only care about dates. Thus, the Examiner's dismissal of the Applicant's argument that "Windows" was not published before the filing date of the patent application is inappropriate.

The Applicant requests that the Examiner explain in detail why the Examiner thinks "Windows" – a document published after the filing date of the patent application – is available as a reference under 35 U.S.C. § 103(a). **Specifically, the Applicant requests that the Examiner identify under which subsection of 35 U.S.C. § 102 the Examiner thinks "Windows" qualifies as prior art, as required by 35 U.S.C. § 103(a), and explain why "Windows" qualifies as prior art under that subsection of 35 U.S.C. § 102.**

The Examiner has not responded to the Applicant's arguments about the statute and the M.P.E.P.

In response to the Office Action dated September 8, 2004, the Applicant argued why, under both the statute and the M.P.E.P., "Windows" is not proper prior art. As the Examiner made no response to the argument that "Windows" is not proper prior art under the M.P.E.P., the Examiner's Office Action is incomplete. For the Examiner's convenience, this argument is restated here:

> Windows is a non-patent document, which the Examiner retrieved from an on-line database. According to M.P.E.P. § 2128, "Prior art disclosures on the Internet or on an on-line database are considered to be publicly available as of the date the item was publicly posted. If the publication does not include a publication date (or retrieval date), it cannot be relied upon as prior art under 35 U.S.C. § 102(a) or (b), although it may be relied upon to provide evidence regarding the state of the art." Windows, however, does not include a publication date, and the Examiner has failed to establish the publication date of Windows. The closest the Examiner has come to establishing a date for Windows is the date the Examiner printed Windows: August 31, 2004. This means that Windows is not prior art under 35 U.S.C. § 102(a) or (b), and therefore is not available as a reference under 35 U.S.C. § 103(a).

The Applicant also includes here the argument presented in response to the Office Action dated April 21, 2005:

> In responding to the Applicant's arguments, the Examiner has indicated that the reference "clearly states the date (year 2000)". The Applicant respectfully disagrees. The reference describes the Windows 2000 operating system, but this does not define a date of publication. "WINDOWS 2000" is merely the name or trademark of the product, and does not establish a date. The only date provided is that on the bottom right-hand corner of the page, which indicates the date the Examiner printed the document: August 31, 2004. If the Examiner wishes to establish that this document was published before that date, the Examiner needs to prove the publication date: an unsupported assertion that the document was published in the year 2000 is insufficient. According to M.P.E.P. 901.06, "[a]ll printed publications may be used as references, the date to be cited being the publication date". The mere fact that the document discusses an object that existed in the year 2000 does not establish the document as having been published in the year 2000.
>
> Further, the Applicant refers the Examiner to M.P.E.P. 2128. In that section, when discussing Electronic Publications as Prior Art, the M.P.E.P. states that "[i]f the publication does not include a publication date (or retrieval date), it cannot be relied upon as prior art under 35 U.S.C. 102(a) or (b), although it may be relied upon to provide evidence regarding the state of the art." As no publication date can be established for the reference, under M.P.E.P. 2128, this reference is not available under 35 U.S.C. § 102(a) or (b), and therefore is not available under 35 U.S.C. § 103(a).
>
> Finally, the best date (of any sort) that can be established for the reference is October 28, 2001. The undersigned visited the URL of the reference (http://www.comptechdoc.org/os/windows/win2k/win2kauthentication.html), and found a set of links on the left side of the website page, which were omitted from the printout provided by the Examiner. . . . Upon selecting the link titled "Introduction", the undersigned was taken to the URL for the introduction to the document which included the reference. This document is entitled "The CTDP

> Windows 2000 Tutorial Version 0.6.1 Oct 28, 2001" (http://www.comptechdoc.org/os/windows/win2k/index.html). It is worth noting on the left column of both website pages, item 64 is titled "Authentication" and is a hyperlink that brings the reader back to the website page of the reference. . . . Given that the version of the introduction to the reference cited by the Examiner is dated October 28, 2001, this is the earliest date that can be assigned to the reference. (It is worth noting that even this date cannot be fixed as a publication date; it is the date on which the entire tutorial was considered complete, but does not provide any definitive date as to when the tutorial was actually made available to the public.) As October 28, 2001 is more than seven months after the filing date of this patent application, even this date fails to establish the reference as prior art.

In the Office Action dated June 9, 2006, the Examiner stated that the Applicant's argument that "Windows" is not valid prior art was "not persuasive because the Office cited the prior art as a teaching regarding the Windows 2000 rather than the prior art being published in the year 2000" (*see* Office Action dated June 9, 2006, page 2). The Examiner's argument ignores both the statute and the M.P.E.P., which both require that prior art, to be proper, be published at least before the filing date of the patent application. Whether or not Microsoft Windows 2000 taught the claimed features (which, as argued above, the Applicant disputes) is not the point: the statute and the M.P.E.P. require that the teaching the Examiner cites be published before the date of the patent application. The reason for this requirement is to ensure that if the patent is denied, it is because the claimed invention was known before the filing date of the patent application.

If the Examiner's reasoning is to be believed, it means that anyone can assert, after the fact, that an invention was previously known. Such an assertion would then be sufficient to invalidate a patent application. The Applicant suggests that this is inappropriate: all such an assertion proves is that someone thinks the claimed subject matter might have been known before the filing date of the patent application. Without corroborating evidence that the claimed subject matter was actually known before the filing date of the application, the mere assertion that the claimed subject matter was previously known should be insufficient.

If the Examiner believes that Microsoft Windows 2000 teaches the claimed invention, the Examiner should find a reference dated before the filing date of the patent application that teaches the claimed subject matter. The Examiner has indicated that "no other prior art of record teaches the claimed subject matter" (*see* Office Action dated June 9, 2006, page 3). The Applicant finds it hard to believe that if Microsoft Windows 2000 taught the claimed subject matter, as the Examiner insists, that there is no reference published before the filing date of the patent application that teaches the claimed subject matter.

As discussed above, the Applicant has conducted a search to determine how Microsoft Windows 2000 performed authentication in its original release, even though the Applicant does not bear the burden of conducting such a search. The Applicant has found no reference published before the filing date of the patent application that describes Microsoft Windows 2000 as including the cited features of "Windows".

"Windows" is hearsay evidence

The Examiner is attempting to use "Windows" for the truth of the matter asserted – namely, that Microsoft Windows 2000 included authentication as described, before the filing date of this patent application (*see* Office Action dated June 9, 2006, page 2 ("the Office cited the prior art as a teaching regarding the Windows 2000 rather than the prior art being published in the year 2000")). This is hearsay. According to Federal Rules of Evidence (FRE) Rule 801(c), "'[h]earsay' is a statement, other than one made by the declarant . . . offered in evidence to prove the truth of the matter asserted". According to FRE 802, "[h]earsay is not admissible except as provided by these rules or by other rules prescribed by the Supreme Court pursuant to statutory authority or by Act of Congress". FRE 803-807 provide a number of exceptions to FRE 802, but none of these exceptions are applicable here. As "Windows" is hearsay evidence and no exception to the hearsay rule permits the entry of "Windows" as evidence, it should not be admitted in considering the patentability of the claimed invention.

The Applicant recognizes that the Examiner might be attempting to argue that the Examiner's rejection is based not on the reference itself, but rather as "a teaching regarding the [Microsoft] Windows 2000" product as of its release date. If that is the case, then the rejection should not be centered around "Windows", but rather on the Microsoft Windows 2000 product. But this does not relieve the Examiner's burden of establishing what was known before the filing

date of the patent application: the Examiner has failed to meet this burden in this case. "Windows" is a statement by its author that its author thinks Microsoft Windows 2000 included such authentication features at some unidentified date before October 28, 2001: "Windows" cannot be read more broadly than this interpretation without becoming hearsay evidence.

The Applicant can only respond to the rejection the Examiner has made, and the Examiner has rejected the claims over "Windows", not over Microsoft Windows 2000. But even if the Examiner is relying indirectly on "Windows" to prove what was in Microsoft Windows 2000, the burden still lies with the Examiner to show that the claimed subject matter was published, known, or used by others before the filing date of the patent application: that is the reason behind the language in 35 U.S.C. §§ 102-103. To use "Windows" in the manner the Examiner is proposing is to offer the reference "for the truth of the matter asserted", which is inadmissible hearsay.

The Examiner is attempting to shift the burden to the Applicant

In the Office Action dated June 9, 2006, the Examiner has said that "Applicant has asserted that the features of Windows 2000 may not have been actually known to others as an actual feature until after the first release. This may be persuasive. Applicant is requested to file an appropriate affidavit stating that this is true" (*see* Office Action dated June 9, 2006, pages 2-3).

The Examiner is improperly attempting to shift to the Applicant the burden of proof of the non-existence of prior art. Aside from the near impossibility of proving that something does not exist, according to the statute, "[a] person shall be entitled to a patent unless. . . ." (*see* 35 U.S.C. § 102). In other words, unless the Examiner can present a prima facie argument as to why an applicant should be denied a patent, an applicant is entitled to the patent. This demonstrates that the burden is supposed to be on the Examiner to show that the applicant is not entitled to the patent. To say that the Applicant needs to show that the features in dispute in this case were not part of the initial release of Microsoft Windows 2000 is to assume that the reference is sufficient despite the fact that the reference was published after the filing date of the patent application, and to require the Applicant to disprove the reference's applicability. This is shifting the burden of proof to the Applicant, which is inappropriate. Such burden-shifting in general would require an Applicant to prove a negative, which is generally impossible to do.

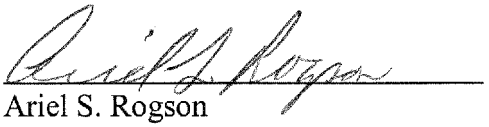
The Examiner bears the burden of showing by prima facie evidence that the claims are properly rejected: the Applicant has, in this case, made sufficient arguments as to why "Windows" is insufficient to make such a rejection. The burden should now be back on the Examiner to show why the rejection can be maintained, if the Examiner wants to maintain the rejection.

As discussed above, despite the fact that the Applicant does not bear the burden of searching for what Microsoft Windows 2000 included in its original release, the Applicant has conducted a search to attempt to define how Microsoft Windows 2000 performed authentication in its original release. The Applicant has found no publications that would qualify as prior art that support the Examiner's contention that the original release of Microsoft Windows 2000 included the features described in "Windows", let alone the claimed invention.

For the foregoing reasons, reconsideration and allowance of claims 1-47 of the application as amended is requested. The Examiner is encouraged to telephone the undersigned at (503) 222-3613 if it appears that an interview would be helpful in advancing the case.

Respectfully submitted,

MARGER JOHNSON & McCOLLOM, P.C.



Ariel S. Rogson
Reg. No. 43,054

MARGER JOHNSON & McCOLLOM, P.C.
210 SW Morrison Street, Suite 400
Portland, OR 97204
503-222-3613
Customer No. 45842

Next Page

The CTDP Windows 2000 Tutorial Version 0.6.1 Oct 28, 2001

This guide may have inaccuracies, use at your own risk.

Introduction

This Windows 2000 tutorial is best used after reading the CTDP Windows NT guides or with the CTDP Windows NT guides in order to fully understand the operation and use of this operating system. Also, to understand Active Directory, the reader should have some knowledge of object oriented concepts. It should be helpful to read the Object Guide and the UML Guide on this website. RFCs are posted at www.ietf.org.

There are four Windows 2000 operating systems:

- Windows 2000 Professional - Supports up to two processors and up to 4GB of RAM. Used as a workstation or client computer and it is the replacement for Windows NT Workstation.
- Windows 2000 Server - Supports up to four processors and up to 4GB of RAM. It is used for web, application, print and file servers.
- Windows 2000 Advanced Server - Supports up to eight processors and up to 8GB of RAM. It is used in an enterprise network and very useful as an SQL server.
- Windows 2000 Datacenter Server - Supports up to 32 processors and up to 64GB of RAM. It is used in an enterprise network to support extremely large databases and real time processing.

System	Microprocessor	RAM	HD Requirements
Windows 2000	Pentium 133	64Mb	650 MB free (2 G recommended)
Windows 2000 Server	Pentium 133	128Mb (256Mb Recommended)	1 GB free (2 G recommended)
Windows 2000 Advanced Server	Pentium 133	256Mb	1 Gb free (2 G recommended)
Windows 2000 Datacenter Server	Pentium 133	256Mb	1 GB free (2 G recommended)

VGA video or better is required for all systems alnog with a CDROM, and keyboard. Also a mouse, floppy disk drive and network card should be on the system, but are not required.

100MB additional disk space may be required if using a FAT file system and over the network installations also require additional hard disk room.

New Features of Windows 2000 over NT

- Plug and play support.
- Keberos 5 security protocol.

- New file systems:
  - FAT32 support - A file allocation table operating system that supports larger disk partition size than older FAT filesystems.
  - EFS - Encrypting File System support.
- Internet Explorer version 5 with XML support and Outlook Express version 5.
- Additional control panel power options.
- Can support up to 10 displays simultaneously.

User interface

The Windows 2000 user interface is similar to Windows 98. Some selections using various icons and selections include:

- Recycle Bin - Used to store deleted files and folders. When emptied, files or folders are gone for good.
- My Network Places Icon
  - Add Network Place selection - Used to connect to a shared network folder or the world wide web.
  - Computers Near Me selection - Used to connect to computers in your domain or workgroup.
  - Entire Network selection
    - Used to view all domains, workgroups, and computers on the organizational network.
    - Used to search for a specific computer.
    - Used to search for specific files or folders.
- Windows Explorer - To run, select "Start", "Programs", "Accessories", and "Windows Explorer".

Platform Support

Windows 2000 will only run on the Intel Pentium platforms. Windows NT additionally supported the Compaq Alpha (previously Dec Alpha) platform, the MIPS R4000, and the Power PC. The Alpha platform was not supported after Windows NT service pack (SP) 6, and the other platforms lost support after Windows NT service pack 1.

Windows 2000 does not allow direct hardware access. All hardware access must be through the hardware abstraction layer (HAL).

Other Support

- **Windows NT 4.0 domains**
- User and group accounts using Windows 2000 Active Directory or a local database.
- **IPSEC** - Internet security protocol.
- **Smart cards.**

Windows
Operating Systems
Home

Basic Overview of Kerberos User Authentication Protocol in Windows 2000

This article was previously published under Q217098

Article ID : 217098
Last Review : November 21, 2003
Revision : 3.0

SUMMARY

This article describes Kerberos user authentication in Windows 2000.

Note that in matters regarding authentication, Windows 2000 is completely backwards compatible. This article focuses on Kerberos user authentication in a pure Windows 2000 environment: authentication between Windows 2000 servers and Windows 2000 clients.

MORE INFORMATION

Windows 2000 provides support for MIT Kerberos v.5 authentication, as defined in IETF RFC 1510. The Kerberos protocol is composed of three subprotocols. The subprotocol in which the KDC (Key Distribution Center) gives the client a logon session key and a TGT (Ticket Granting Ticket) is called the Authentication Service (AS) Exchange. The subprotocol in which the KDC distributes a service session key and a ticket for the service is called the Ticket-Granting Service (TGS) Exchange. The subprotocol in which the client pre-sends the ticket for admission to a service is called the Client/server (CS) Exchange.

The following is a simple overview of the chain of communication involved in a Kerberos authentication session, between a client workstation and a resource server.

1. Authentication Service (AS) Exchange
   a. User A, at a Microsoft Windows 2000 Professional workstation, logs on to a Microsoft Windows 2000 network, typing her user name and password. The Kerberos client running on A's workstation converts her password to an encryption key, and saves the result in a program variable.
   b. The Kerberos client sends a message to the Key Distribution Server (KDC), of type KRB_AS_REQ (Kerberos Authentication Server Request). This message has two parts:
      - An identification of the user, A, and the service for which she is requesting credentials, the TGS (Ticket-Granting Service)
      - Pre-authentication data, intended to prove that A knows her password. This is simply an authenticator encrypted with A's master key. The master key is generated by running A's password through a OWF.
   c. The KDC, upon receipt of KRB_AS_REQ from A, looks up the user A in its database (the Active Directory), gets her master key, decrypts the pre-authentication data, and evaluates the time stamp inside. If the time stamp passes the test, the KDC can be assured that the pre-authentication data was encrypted with A's master key, and is not merely a captured replay.
   d. Finally, once the KDC has verified A's identity, it will create credentials that the client program on her workstation can present to the Ticket Granting Service (TGS). The credentials are created and deployed as follows...
      - A brand new logon session key, encrypted with A's master key
      - A second copy of the logon session key and A's authorization data, in a Ticket Granting Ticket (TGT), encrypted with the KDC's own master key.
      - Next, the KDC sends these credentials back to the client by replying with a message of type KRB_AS_REP (Kerberos Authentication Response)
      - When the client receives the reply, it decrypts the logon session key via application of A's master key. The session key is then stored in the client workstation's ticket cache. The TGT is extracted from the message, and stored in the cache as well
2. Ticket-Granting Service (TGS) Exchange
   a. At this stage, the Kerberos client running on A's workstation is going to actually request credentials to access the target server, user B, by sending a message of type KRB_TGS_REQ (Kerberos Ticket-Granting Service Request), to the KDC. This message consists of the following components...
      - Identity of the target service for which the client is requesting credentials
      - Authenticator encrypted with the user's logon session key
      - TGT acquired from the AS Exchange

b. The KDC decrypts the TGT with its master key, and extracts A's logon session key. A's logon session key is used to decrypt A's authenticator. If A's authenticator passes the test, the KDC invents a new session key for A to share with B. Two copies of this new session key are sent back to A in a single message, encrypted as follows..
   - One copy is encrypted using A's logon session key
   - The second copy is encrypted using the target server's master key, in a ticket along with A's authorization data.

c. A decrypts the target server session key, using her logon session key, and stores the session key in her cache, along with the target server ticket.

3. Client-Server (CS) Exchange
   a. A's Kerberos client is now ready to be authenticated by the target server, B. A's client sends B a message of type KRB_AP_REQ (Kerberos Application Request). This message contains:
      - An authenticator encrypted with the session key for B
      - The ticket for sessions with B, encrypted with B's master key
      - A flag indicating whether the client requests mutual authentication.

   b. B decrypts the ticket, and extracts A's authorization data and session key. B uses the session key to decrypt A's authenticator, and evaluates the time stamp. If the authenticator passes the test, B looks for a mutual authentication flag. If this flag is set, B uses the session key to encrypt the time from A's authenticator, and returns the result to A in a message of type KRB_AP_REP (Kerberos Application Reply)

   c. A decrypts the reply with the session key. If the authenticator is identical to the one that she sent B, the client is assured that the server is genuine, and the connection proceeds.

APPLIES TO

- Microsoft Windows 2000 Server
- Microsoft Windows 2000 Advanced Server
- Microsoft Windows 2000 Professional Edition
- Microsoft Windows 2000 Datacenter Server

Keywords: kbinfo kbnetwork KB217098

List of Security Fixes in Windows 2000 Service Pack 3

This article was previously published under Q324953

Article ID : 324953
Last Review : March 9, 2006
Revision : 4.0

SUMMARY

Windows 2000 Service Pack 3 (SP3) fixes the following security problems or adds the following updates:

- **Default Diffie-Hellman SChannel Certificate Selection on Web Enrollment Page Causes Error: 0x80090008 - NTE_BAD_ALGID**

  The default Diffie-Hellman (DH) certificate selection of **Both** on the Certificate Authority (CA) Web enrollment page results in Error: 0x80090008 - NTE_BAD_ALGID. An update is available to change the default certificate selection to **Signature** on Windows 2000 CAs.
- **Set LAN Manager (LM) One-Way Function (OWF) Password Results in Access Denied Error**

  When you use the **UserAllInformation** level to set passwords, you are successful if you pass password information in clear text, but if you try to use LM OWF, you receive an Access Denied error (error code 5).
- **Invalid Entry in Certificate Store Causes Event ID 1008**

  An update to the CertOpenStore function allows it to skip an invalid entry.
- **Incorrect Key Usage with Encrypting File System (EFS) May Cause Access Violation in LSASS.exe**

  For EFS to use other certificates in Windows 2000, it is updated to look for the EFS object identifier (also known as OID) in the enhanced key usage (CERT_ENHKEY_USAGE) structure in the certificate store.
- **Imported Certificate More Than 512-Bit Is Considered Invalid for Encrypting File System (EFS)**

  This update imports certificates up to 1024-bit by using Base cryptographic service provider (CSP) instead of Enhanced CSP.
- **Roaming User Cannot Delete a User Key Container That Was Created on Different Computer**

  The code in the DeleteContainerInfo function is updated to allow it to delete a container, by the name it was given, that was created on a different computer.
- **New Key Distribution Center (KDC) Certificate Is Not Used After Enrollment**

  When a new KDC certificate is obtained, after the previous one expires, Windows continues to use the expired certificate until the server is restarted, or the KDC service is restarted.
- **Buffer Overrun Vulnerability in the Runas Command**

  When you supply a program name that is about 600 characters in length in the **Runas** command line, you may receive a memory access error. This might allow the execution of malicious code, or be used as a denial of service attack.
- **File Decryption Following a Password Change May Be Unsuccessful in Domains with Both Windows 2000 and Windows Server 2003 Domain Controllers**

  After a user account password change, the Data Protection API (DAPI) contacts the domain controller to have the domain controller decrypt the "master key." Because of a change in the encryption scheme in the Windows Server 2003 family, if the master key was encrypted by a Windows Server 2003 domain controller, an attempt to decrypt it by using a Windows 2000 domain controller is unsuccessful.
- **Certificate Is Not Removed from the Certification Authority Store After Removing It from the Encrypting File System (EFS)**

  This update removes a certificate and the certificates in its chain of certificates (if they are not in the chain of other remaining certificates in the EFS store) from the certification authority when the certificate is removed from the EFS store.
- **User Credentials Remain in Memory Buffer After Using the Runas Command**

  After using the Runas command-line utility, a user's credentials are not erased after quitting the program. To exploit this vulnerability, a malicious user must have interactive access to the computer. A program might wait for a RunAs session to quit, and then subsequently search for that user's credentials.
- **Malicious Code That Listens on the Same Pipe as the RunAs Service Might Receive User Credentials**

  This update prevents the Runas command from running if the RunAs service is stopped.
- **Possible Denial of Service Vulnerability in the Windows 2000 RunAs Service**

  If you disable the named pipe on which the RunAs service listens, the secondary logon function (Runas) is effectively disabled. Malicious code that is run with administrative privileges might be used to block activity on this pipe. This update to the RunAs service permits multiple instances of this pipe, and holds state data for each client.
- **Buffer Overflow Vulnerability in Telnet.exe**

  Passing 252 characters as the port parameter in the Telnet.exe command line results in a buffer overflow. This may allow malicious code to run in the context of the currently logged-on user.
- **Kerberos Change Password Is Unsuccessful in a MIT Realm When the Principal Requires Pre-authorization**

  The Kerberos.dll file is updated to make sure that the KerbLookupMitRealm function is always called.

- **Links Can Contain Encoded Text That Can Add HTTP Request Headers**

  This update includes an updated Wininet.dll file that checks host names for invalid characters and returns an error if it finds any.

- **Vulnerability in the Unsafe ActiveX Control Dialog Box**

  The Internet Explorer dialog box that prompts you to confirm the running of an unsafe ActiveX control can be hidden by covering it with a chromeless window. This may trick a user into accepting the installation of an unsafe ActiveX control.

- **Renaming a Computer or Joining a Computer to the Domain**

  This update removes the need for Inheritable Access Control Entries to rename a computer or to join a computer to the domain.

  This update also fixes the problem described in the following Microsoft Knowledge Base (KB) article:

  290533 (http://support.microsoft.com/kb/290533/EN-US/) User Permission to Add Workstation to Domain Includes Permission to Rename Computer Account

- **Group Policy Object Version Number Changes to 0 (Zero) After 65535 Changes**

  A Group Policy object (GPO) with a version number of zero is determined to be a newly created blank GPO. The Group Policy engine uses this version number to determine whether to apply the GPO (a version zero GPO is skipped). When you change a GPO with a version number of 65535, it is assigned a version number of zero, causing it to be skipped by the Group Policy engine.

- **Denial of Service Vulnerability in the Internet Key Exchange Service**

  A denial of service attack can be carried out against Windows 2000 computers that run Internet Key Exchange (IKE) by flooding them with User Datagram Protocol (UDP) packets.

- **Unsigned Webview Templates**

  This update includes an updated security policy that prevents unsigned webview templates from running.

- **Security-Related Problems in Microsoft Internet Explorer**

  This update prevents the reading of a user's files by using a script. The update also includes the fixes that are described in the following Microsoft Knowledge Base (KB) articles:

  317745 (http://support.microsoft.com/kb/317745/EN-US/) MS02-005: Patch Is Available for File Download Dialog Box Spoofing Vulnerability

  312461 (http://support.microsoft.com/kb/312461/EN-US/) MS01-055: Internet Explorer Cookie Data Can Be Exposed or Altered Through Script Injection

  282062 (http://support.microsoft.com/kb/282062/EN-US/) IIS Does Not Authenticate for the /_AuthChangeUrl URL

  317727 (http://support.microsoft.com/kb/317727/EN-US/) MS02-005: Patch Is Available for the Application Invocation via Content-Type Field Vulnerability

- **Security Audit Is Not Performed When You Add Users from Another Domain to Universal Groups**

  Auditing is not performed when you add users to, or remove users from a universal group, when those users are from a different domain in the same forest.

- **Improved or Updated Security in the Internet Key Exchange Process**

  This security update prevents a man-in-the-middle attack from being performed in the Internet Key Exchange process. This update causes the Windows 2000 IPSEC initiator and responder to validate the Internet Key Exchange (IKE) Main Mode HASH.

- **Unauthorized DHCP Server Message Block Server Collects NTLM Hashes**

  A Windows 2000 Server Message Block server might be created that sends a null challenge and therefore receives a user's NT LAN Manager hash (challenge/response pairs).

- **"Fail Privilege Use" Audit Entry Is Not Generated**

  An audit entry is not generated when users without proper permissions try to view the security log. This update adds a return code check that meets the Common Criteria Security evaluation (C2) requirement.

- **IPSEC Driver Drops Certain Packet Fragments**

  Fragmented IPSEC packet fragments of a certain size are dropped.

- **Nonsecure Communication Is Accepted When the 'Accept Unsecured Communication' Option Is Not Selected**

  IPSec accepts nonsecure packets when the **Accept unsecured communication** check box in the IPSec filter is not selected but the **Fall back to unsecured communication with non IPSec-aware computer** is selected.

- **Mounting a Volume to a Folder on the Same Volume Causes Windows Explorer to Stop Responding**

  When you try to edit the security permission of a volume that is looped to itself (a volume that is mounted to a folder on the same volume), the program from which you try to apply the security permissions stops responding (crashes).

- **GetEffectiveRightsFromAcl() Function Returns Incorrect Access Mask**

  After you install Service Pack 2 (SP2) for Windows 2000, the GetEffectiveRightsFromAcl() function no longer returns the correct 32-bit value that specifies the rights that are permitted or denied in an access control entry (ACE).

- **Incorrect Location Checked When Verifying Whether an Audit Category Is Enabled**

  The LsaIWriteAuditEvent function checks the wrong category when it verifies that auditing is enabled for a category.

- **The Close Object Audit Entry Does Not Use a Non-System Account**

  Before installing this update, the Open Object audit entry runs under the account of the current user but the Close Object audit entry is generated by using the SYSTEM account.

- **Flooding Port 464 on a Domain Controller Causes "Spike" in CPU Usage and Memory Leak**

  Repeatedly running a script or program that floods port 464 with hundreds of connections may cause the Local Security Authority (LSA) to consume about 90 percent CPU usage. Also, LSA memory usage increases by about 10 megabytes (MB). After this attempted denial of service attack, CPU usage remains at the high level for about 45 minutes before it returns to typical levels.

- **Strong Password Function Does Not Recognize the Forward Slash Character as a Special Character**

  This update changes the strong password dynamic link library file (Passfilt.dll) to have it recognize the forward slash (**/**) character as a "Special Character" in strong password creation.

- **Private Key Persists in Memory**

  Two copies of a user's private key remain in memory and persist even when the user logs off the computer.

- **Signing and Encrypting of Messages with NT LAN Manager (NTLM)**

  This update supports the signing and encrypting of messages with NTLM.

- **Remote Procedure Call with Invalid Parameters Causes Error in Netdde.exe**

  When a remote procedure call (RPC) passes invalid parameters to **\pipe\nddeapi**, the NETDDE server may incorrectly filter these invalid parameters. As a result, you may receive the following error message:

  > **NETDDE.EXE has generated errors and will be closed by Windows. You will have to restart the program. An error log is being created.**

- **Buffer Overrun Vulnerability Exists in the Dynamic Host Configuration Protocol (DHCP) Service**

  An unchecked buffer exists in the DHCP service that can be remotely accessed through a named pipe that does not provide enough access control. This exploit might permit malicious code to run in the context of the SYSTEM account.

- **Incorrect DHCP Security Access Mask**

  This update changes the Dynamic Host Configuration Protocol (DHCP) security access mask to restrict user permissions to View and Read permissions.

- **Access Violation in Terminal Services License Manager**

  An access violation (AV) occurs in License Manager (Licmgr.exe) when you try to refresh the server settings, and the connection to the target licensing server has been lost.

MORE INFORMATION

305330 (http://support.microsoft.com/kb/305330/EN-US/) Trusting Domain May Allow Privilege Elevation from Trusted Domain User

APPLIES TO

- Microsoft Windows 2000 Service Pack 3
- Microsoft Windows 2000 Advanced Server
- Microsoft Windows 2000 Service Pack 3

Keywords: kbinfo kbwin2000sp3fix kbsecurity kbpending KB324953

one-way function

<*cryptography*, *mathematics*> A function which is easy to compute but whose inverse is very difficult to compute. Such functions have important applications in cryptography, specifically in public-key cryptography.

See also: trapdoor function.

(2001-05-10)

Try this search on Wikipedia, OneLook, Google

Nearby terms: ones complement « One-Time Password « One Time Programmable Read-Only Memory « **one-way function** » one-way hash function » on-line » On-Line Analytical Processing